Development of versatile online monitoring for EUDAQ2

Task 3.4: Development of DAQ software for next generation beam tests

Andreas Loeschcke Centeno
a.loeschcke-centeno@sussex.ac.uk

Fabrizio Salvatore, Lennart Huth

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004761.

Task -> Milestone -> Deliverable

Tasks

Task #	Task name	Task Leader
3.4	Development of DAQ software for next generation beam tests	Lennart Huth (DESY)

Milestones

MS #	Milestone name	Lead beneficiary	Due Date (in months)	Means of verification
MS10	Monitoring software developed	39 - UCL	30	Use in beam tests (Task 3.4)

Deliverables

D #	Deliverable name	Lead beneficiary	Type	Due Date (in months)
D3.4	New software developments available for use	39 - UCL	Report	39

- Versatile: any detector / any test beam setup
- within EUDAQ2 framework

- Development at advanced stage
- Test cases available

- Wide release into repository
- Ideally some use-cases

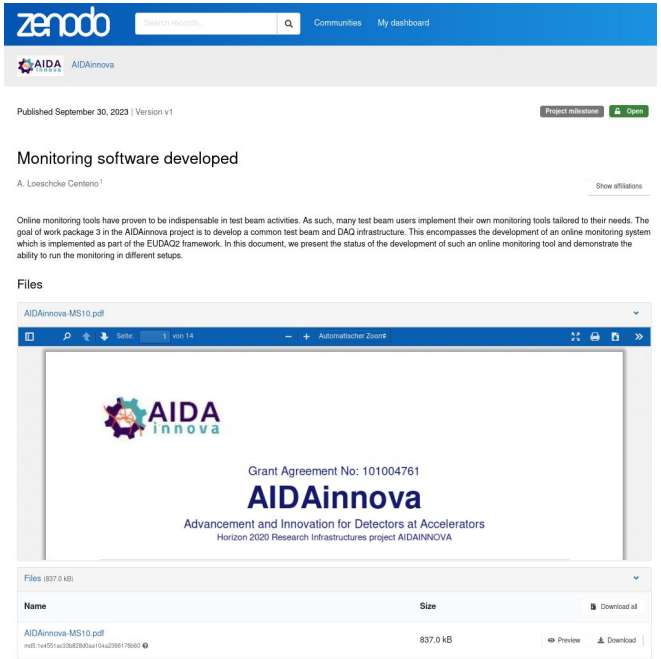
Task -> Milestone -> Deliverable

Milestones

Year 3

MS no.	Milestone name	WP	Task	Planned delivery date	Actual delivery date	Status	Comments
MS10	Monitoring software developed	WP3	3.4	M30	30/09/2023	Achieved	Report

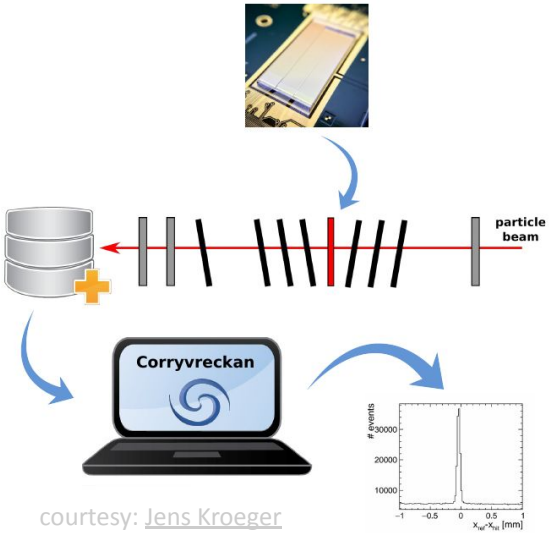
- Link to Report
- Last status update: AIDAinnova 2nd Annual Meeting (26.04.2023)



Last Year's Meeting Status/Recap: CorryMonitor

courtesy: Lennart Huth

- Existing EUDAQ2 monitoring not suitable to fulfil requirements
- No need to start from scratch:
  Widely used test beam reconstruction and analysis software corryvreckan is compatible with EUDAQ2 and comes with monitoring capabilities
- Only need to figure out how to seamlessly integrate into EUDAQ2: **CorryMonitor**



courtesy: Jens Kroeger

Last Year's Meeting Status/Recap: CorryMonitor

Corryvreckan:

- Loads EUDAQ2 raw events (requires eudaq::StdEventConverter)
- Setup geometry through .geo file
  - Nomenclature laid out for pixel sensors (pixels, planes)
  - Can be used in any generic way
- Can do hitmaps, correlation between planes, tracking, alignment
  - Load dedicated corryvreckan modules in corry configuration file
- Extensive documentation

corryvreckan .conf file

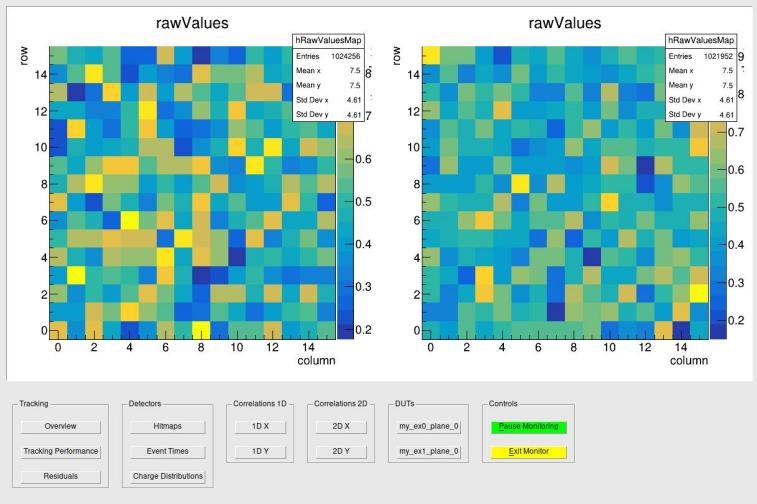
```
[Corryvreckan]
detectors_file = "corrygeo.geo"
detectors_file_updated = "corrygeo_updated.geo"
histogram_file = "corry_histo_file_example.root"

[Metronome]
triggers = 1
event_length = 1s

[EventLoaderEUDAQ2]
type = "Ex0Raw"
file_name = placeholder0.raw
eudaq_loglevel=INFO
buffer_depth=5
inclusive=1

[OnlineMonitor]
dut_plots = [["EventLoaderEUDAQ2/%DUT%/hRawValuesMap", "colz"],
             ["EventLoaderEUDAQ2/%DUT%/hPixelRawValues", "log"]]
hitmaps = [["EventLoaderEUDAQ2/%DUT%/hRawValuesMap", "colz"]]
```

corrygeo.geo

```
[my_ex0_plane_0]
number_of_pixels = 16, 16
orientation = 0deg,0deg,0deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = 0um,0um,0um
role = "dut","reference"
time_resolution = -1ns
type = "ex0raw"

[my_ex0_plane_1]
number_of_pixels = 16, 16
orientation = 0deg,0deg,0deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = 0um,0um, 50m
role = "dut"
time_resolution = -1ns
type = "ex0raw"
```

Last Year's Meeting Status/Recap: CorryMonitor

1. Demonstrated to start and control corryvreckan from EUDAQ2 run control
2. Correctly identifying data files to pass to corryvreckan at runtime with minimal user input for user convenience
3. Showcased a lab setup in which functionality was tested



How to CorryMonitor

1. Startup

```
#!/usr/bin/env sh
BINPATH=../../../bin
$BINPATH/euRun &
sleep 1
$BINPATH/euLog &
sleep 1
$BINPATH/euCliMonitor -n CorryMonitor -t my_mon &
$BINPATH/euCliCollector -n Ex0TgDataCollector -t my_dc0 &
$BINPATH/euCliProducer -n Ex0Producer -t my_pd0 &
```

2. EUDAQ2 .ini file

```
[Monitor.my_mon]
CORRY_PATH = /path/to/bin/corry #executable
```

3. EUDAQ2 .conf file

```
[Monitor.my_mon]
CORRY_CONFIG_PATH=./corryconfig.conf
CORRY_OPTIONS=-v INFO
DATACOLLECTORS_TO_MONITOR = my_dc0
CORRESPONDING_EVENTLOADER_TYPES = Ex0raw
```

corryconfig.conf

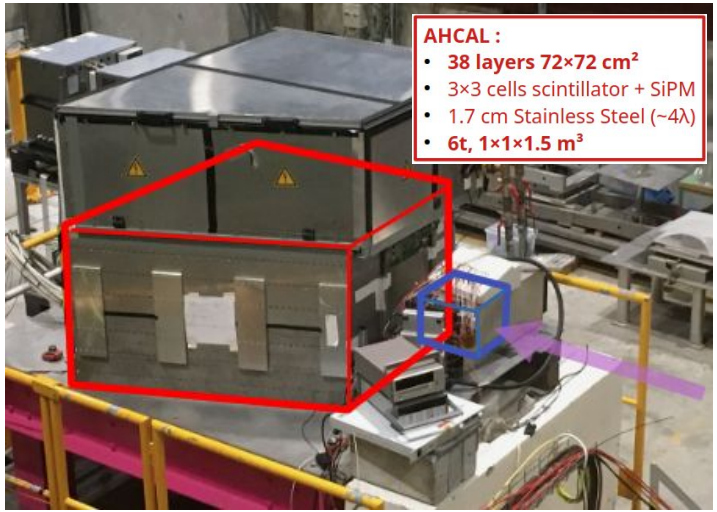
```
[Corryvreckan]
detectors_file = "corrygeo.geo"
detectors_file_updated = "corrygeo_updated.geo"
histogram_file = "corry_histo_file_example.root"

[Metronome]
triggers = 1
event_length = 100s

[EventLoaderEUDAQ2]
type = "Ex0Raw"
file_name = placeholder0.raw
eudaq_loglevel=INFO
buffer_depth=5
inclusive=1

[OnlineMonitor]
dut_plots = [["EventLoaderEUDAQ2/%DUT%/hRawValuesMap", "colz"],
             ["EventLoaderEUDAQ2/%DUT%/hPixelRawValues", "log"]]
hitmaps = [["EventLoaderEUDAQ2/%DUT%/hRawValuesMap", "colz"]]
```

corrygeo.geo

```
[my_ex0_plane_0]
number_of_pixels = 16, 16
orientation = 0deg,0deg,0deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = 0um,0um,0um
role = "dut","reference"
time_resolution = -1ns
type = "ex0raw"
```

AHCAL Monitoring

- After last year's meeting: approached by Jiří Kvasnička
  - Offered CALICE AHCAL **test beam data** to test monitoring with
  - Provided with AHCAL Reader to emulate data taking



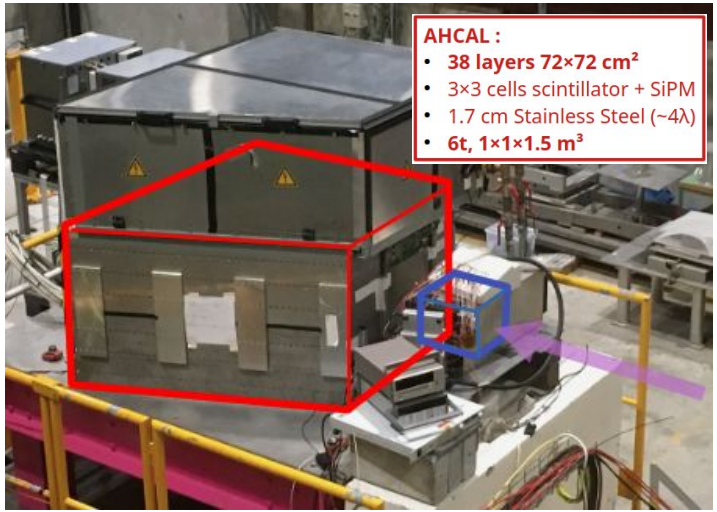
courtesy: Vincent Boudry

corrygeo.geo

```
[AHCAL_0]
material_budget=0.1
number_of_pixels = 24,24
orientation_mode = "xyz"
pixel_pitch = 30.15mm,30.15mm
position=0,0,1746mm
spatial_resolution = 31mm,31mm
role=dut, reference
type="CaliceObject"
time_resolution = 230us

[AHCAL_1]
material_budget=0.1
number_of_pixels = 24,24
orientation_mode = "xyz"
pixel_pitch = 30.15mm,30.15mm
position=0,0,1766mm
spatial_resolution = 31mm,31mm
role=dut
type="CaliceObject"
time_resolution = 230us

[AHCAL_2]
material_budget=0.1
number_of_pixels = 24,24
orientation_mode = "xyz"
pixel_pitch = 30.15mm,30.15mm
position=0,0,1786mm
spatial_resolution = 31mm,31mm
```

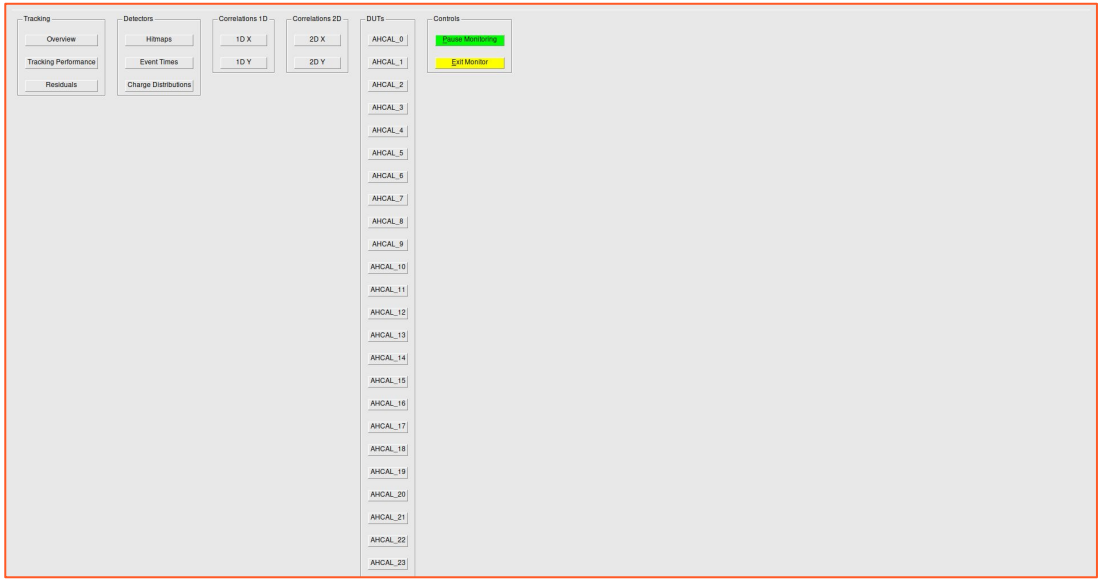
AHCAL Monitoring

- After last year's meeting: approached by Jiří Kvasnička
  - Offered CALICE AHCAL **test beam data** to test monitoring with
  - Provided with AHCAL Reader to emulate data taking



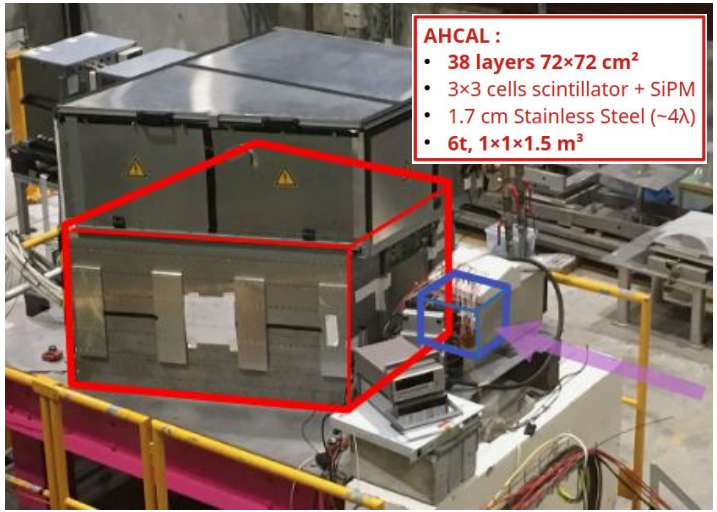
courtesy: Vincent Boudry

- This is the full window, not enough space for plots
- Not even enough space to fit all layers



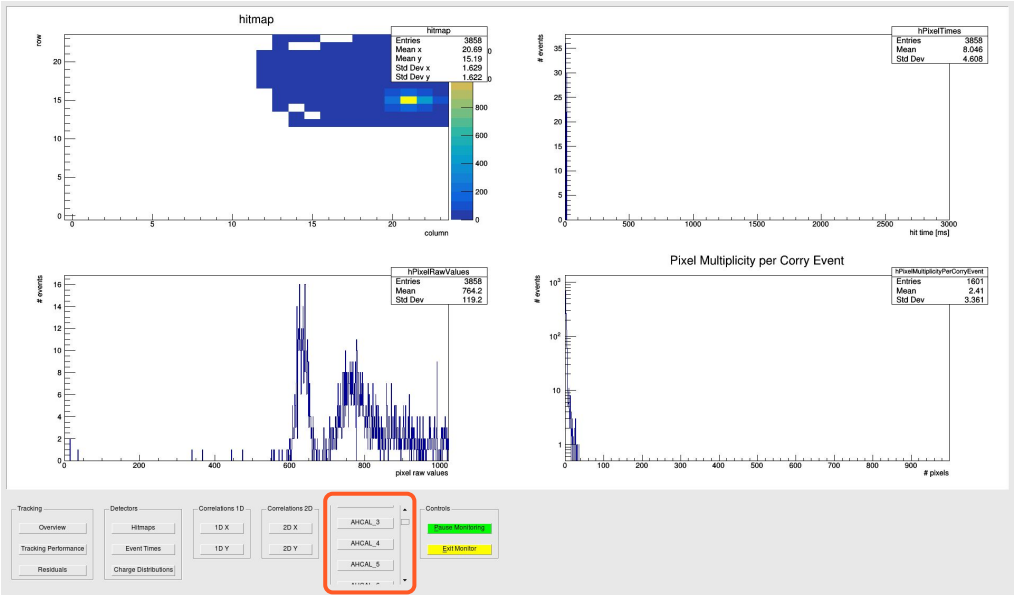
AHCAL Monitoring

- After last year's meeting: approached by Jiří Kvasnička
  - Offered CALICE AHCAL **test beam data** to test monitoring with
  - Provided with AHCAL Reader to emulate data taking



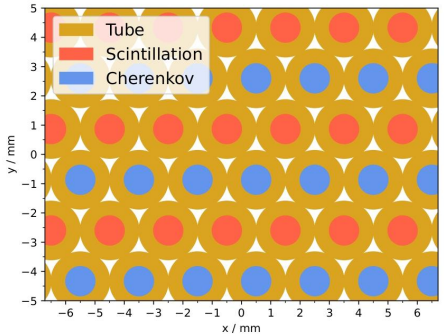
courtesy: Vincent Boudry

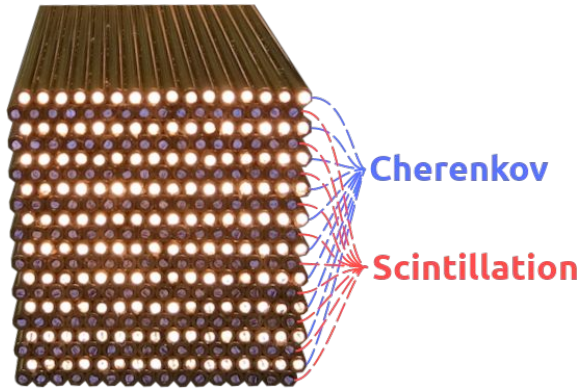
- Implemented scroll bar
- **CorryMonitor works well!**



Dual-Readout Monitoring

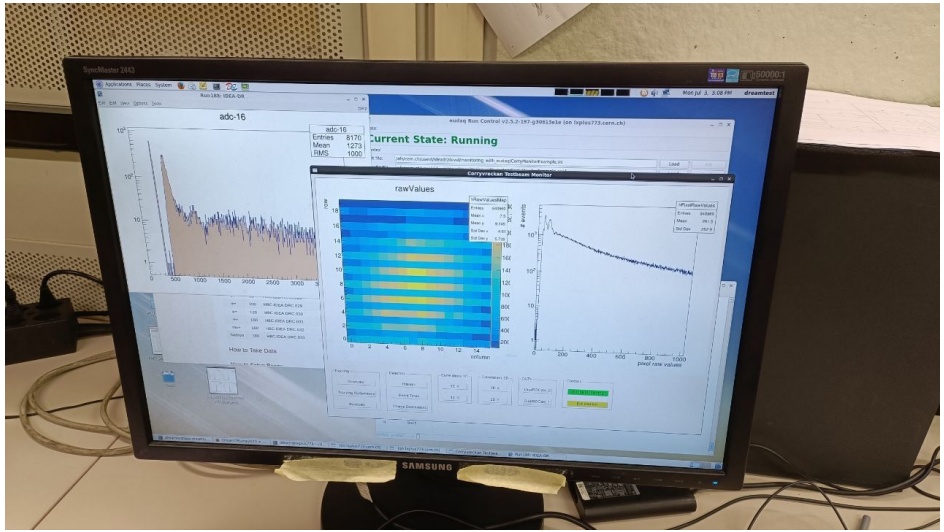
- Test beam campaign in June/July 2023 at SPS North Area
- Dual-readout EM prototype
  - 160 Scintillation, 160 Cherenkov individually read out by SiPMs
  - No longitudinal segmentation, i.e. no “layers”
  - But could define Scintillation and Cherenkov channels as own layers or high gain and low gain

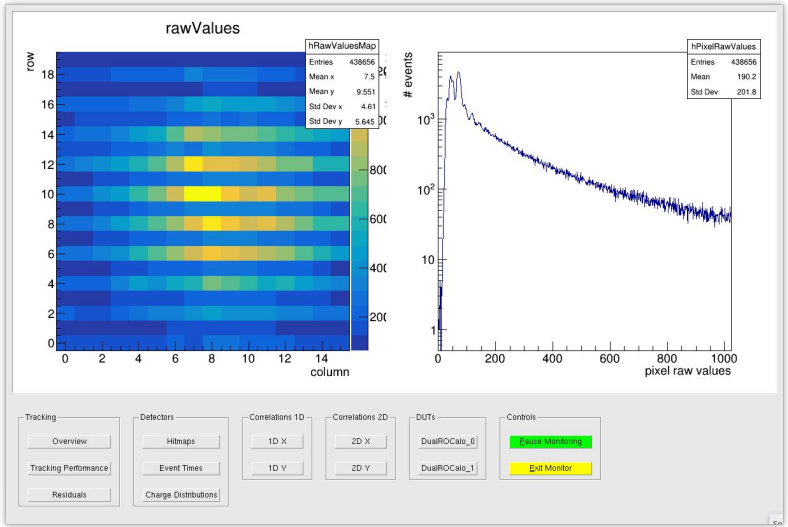




courtesy: INFN-PV Lab

Dual-Readout Monitoring



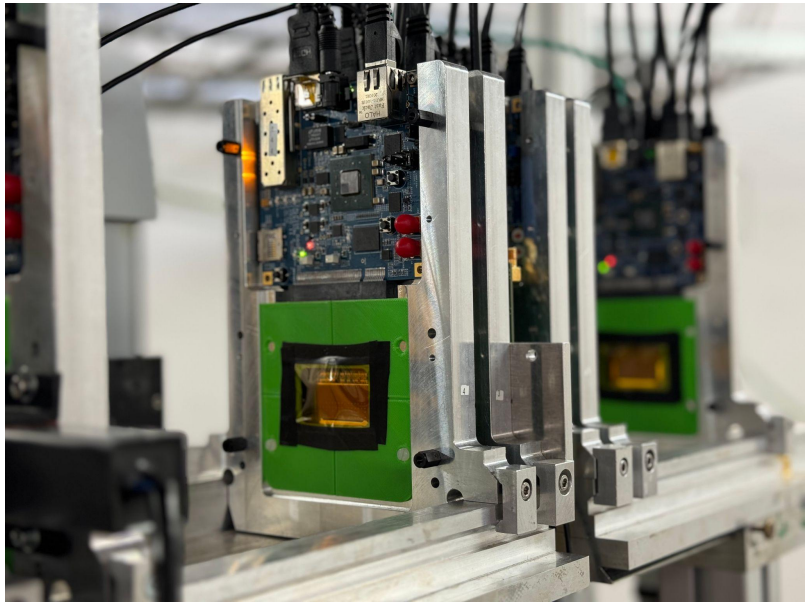


- Again CorryMonitor works as intended!

- Caveat: EUDAQ2 was not used as DAQ control system
- Reading data from disk with dedicated Producer: emulating data taking

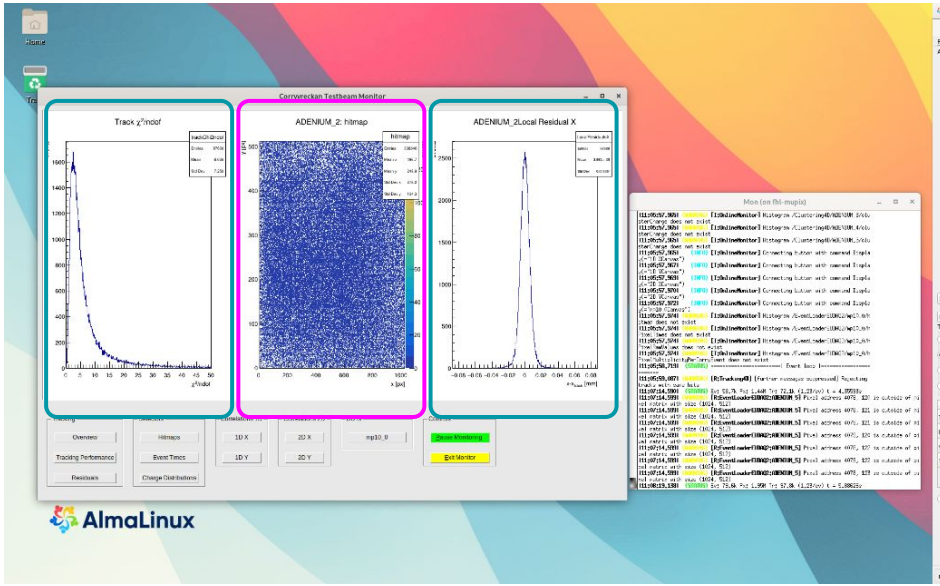
On towards the **Deliverable**

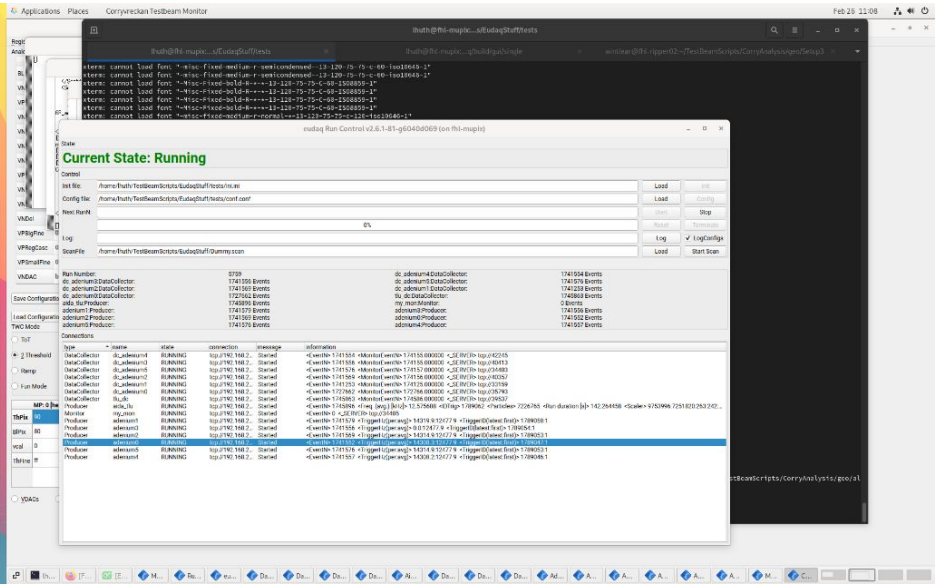
- **Another test beam for testing CorryMonitor!**
- This time with EUDAQ2 as DAQ control system
- TelePix2 beam test with ALPIDE telescope at DESY by Lennart Huth et. al.
- 3 weeks of test beam left plenty of time for testing and providing real-time feedback
- Beam telescope allows to fully exploit corryvreckan functionality (correlations, tracking)



courtesy: Lennart Huth

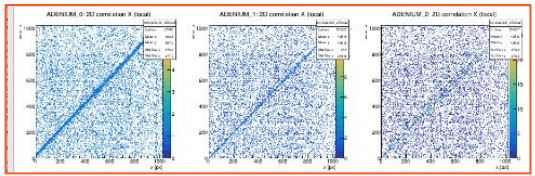
TelePix2 + ALPIDE Monitoring

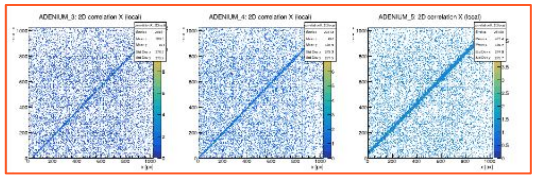




courtesy: Lennart Huth

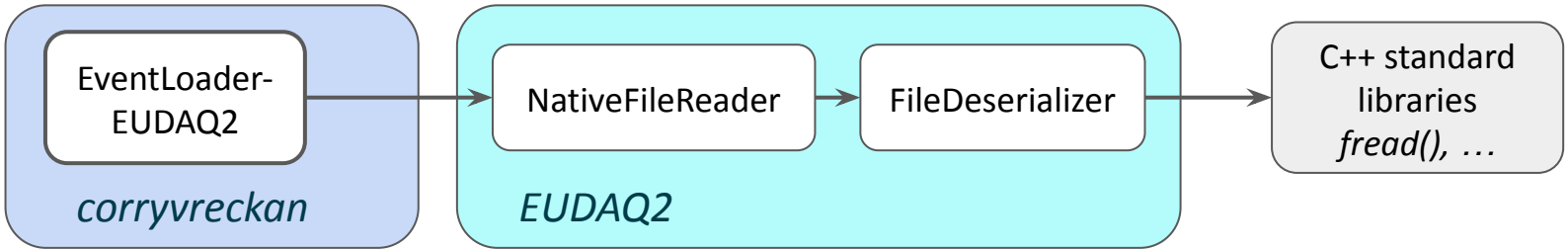
- Hitmaps
- Tracking
- Correlations





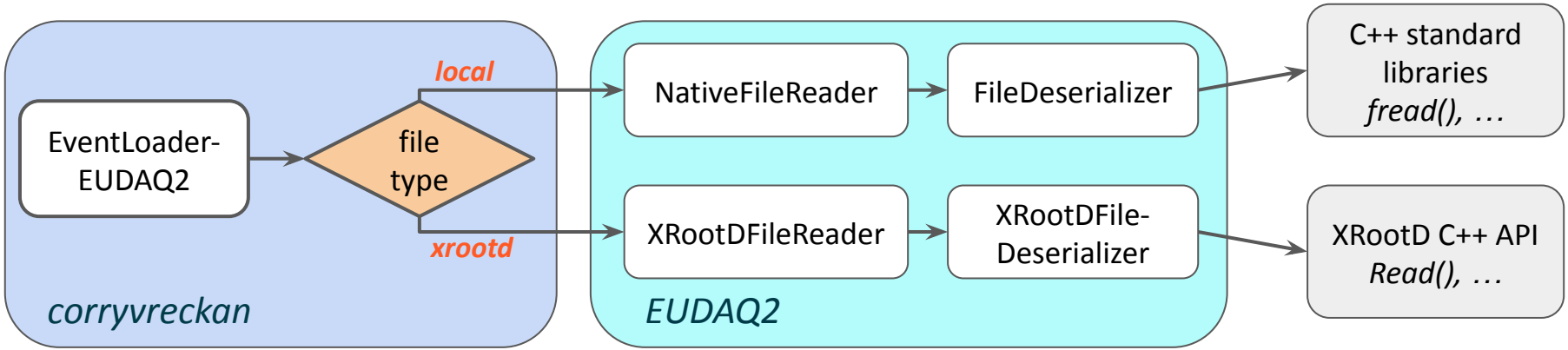
More Features: XRootD

- Want to be able to **read files on different machine**
- Copying files over is not an option
- Need quick way to establish connection
- XRootD software offers solution
  - Allows fast, low latency and scalable data access
- Need to make EUDAQ2 and corryvreckan compatible with XRootD



More Features: XRootD

- Want to be able to **read files on different machine**
- Copying files over is not an option
- Need quick way to establish connection
- XRootD software offers solution
  - Allows fast, low latency and scalable data access
- Need to make EUDAQ2 and corryvreckan compatible with XRootD



More Features: XRootD

- User needs to start XRootD server/ensure connection
- In the .conf file pass server addresses in order (remote DataCollectors first, then local)

```
[Monitor.my_mon]
CORRY_CONFIG_PATH=corryconfig.conf
CORRY_OPTIONS=-v INFO
DATACOLLECTORS_TO_MONITOR = my_xrootd_dc0, my_xrootd_dc1, my_local_dc0
CORRESPONDING_EVENTLOADER_TYPES = Ex0raw, Ex1raw, Ex2raw
XROOTD_ADDRESSES = server0name@127.0.0.1:51234, server1name@127.0.0.1:54321
```

xrootd server-name | IP address | port

- Tested in lab setting
- Unfortunately no opportunity to test in test beam setting so far

Conclusion

- Development well under way
- Reached Milestone
- Had several good testing opportunities
- XRootD functionality needs more testing

- **Well on track for the deliverable!**
  - Due month 39 (June 2024)
  - Started writing up first draft of report
- Presenting CorryMonitor at BTTB12 in April

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004761.

Thank you for your Attention!